

REMARKS

In summary, claims 1-39 are pending. Claims 1-5, 7-18, 20-26, and 28-38 are rejected under 35 U.S.C. § 102. Claims 1, 13, 22, and 34 are herein amended. Claims 2, 6, 19, 23, 27, and 39 are canceled. No new matter is added.

Background

When developing software, it is not uncommon for the developer, or programmer, to want to know if the software will perform as expected. More often than not, the developer wants to know if the software has any errors at various stages in the development of the software, rather than wait until the development is complete. Determining if the software contains any errors, or bugs, is referred to as debugging. Typical software development tools, such as rapid application development (RAD) tools, include the ability to debug the software under development. For example, utilizing Microsoft's VISUAL STUDIO .NET or Visual Basic .NET development tool, an application under development can be debugged by simply depressing the F5 key. Typical RAD programmers use the debugging functionality often. Many developers tend to follow the philosophy of "try it out until it works." That is, they write a section of code and invoke the debugger. If no bugs are found, they write the next section of code. If bugs are found, they fix the code and run the debugger again. Given the frequency with which typical developers start the debugger, it is clear that a quick experience starting the debugger is very important to a developer. Also, fast debugging allows the developer to develop an application quickly, thus reducing development costs.

Applicant's claimed invention is directed to fast debugging. To achieve fast debugging, specific functions related to debugging are performed in advance of the user invoking the debugger. Prior to invoking the debugger: (1) a hosting process, which creates an environment in which the application can be debugged, is started; (2) a runtime environment is loaded in the hosting process; (3) an application domain is created; (4) assemblies are preloaded, and (5) the debugger is attached to the hosting process. Thus, from the user's perspective the time to start debugging is greatly reduced because many of the functions associated with starting the debugger have already been completed when the user invokes the debugger.

DOCKET NO.: MSFT-1757
Application No.: 10/626,958
Office Action Dated: March 30, 2007
Advisory Action Dated: July 19, 20007

PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.114

Claim Rejections - 35 U.S.C. §102

Claims 1-5, 7-18, 20-26, and 28-38 are rejected under 35 U.S.C. §102(a) as being anticipated by an article entitled “Introduction to Series 60 Applications for C++ Developers,” Version 1.0, Nokia corporation, August 2002 (hereinafter referred to as “Series 60 for C++”). Claims 1, 13, 22, and 34 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,353,923, issued to Bogle *et al.* (hereinafter referred to as “Bogle”).

Neither Series 60 for C++ nor Bogle, whether considered together or separately, perform the functions of creating a hosting process, starting a runtime environment in the hosting process, attaching a debugger to the hosting process, and preloading assemblies, before invoking the debugger, as claimed.

Series 60 C++ nowhere teaches the sequence of functions performed prior to invoking a debugger as claimed. In previous Office Actions portions of Series 60 C++ have been cited to reject individual clauses of the claims, but Series 60 C++ does not teach, and the previous Office Actions have not shown, the sequence of performance of functions prior to invoking the debugger as claimed.

Similarly, Bogle nowhere teaches the sequence of functions performed prior to invoking a debugger as claimed. Bogle is directed to an active debugging environment for debugging mixed language scripting code. The first step of Bogle debugging process is to activate the debugger, as illustrated in Figure 5 and column 12, lines 43-46: “The active debugging environment operational steps 500 begin at step 508 ...” Thus, active debugging is invoked at the beginning of the flow diagram of Figure 5 of Bogle. This argument was presented in a previous Office Action Response (Response dated May 30, 2007), but is not addressed in the instant Advisory Action. It appears the flow diagrams of Bogle are being misinterpreted.

Because neither Series 60 for C++ nor Bogle, whether considered together or separately, neither disclose nor suggest, performing the functions of creating a hosting

DOCKET NO.: MSFT-1757
Application No.: 10/626,958
Office Action Dated: March 30, 2007
Advisory Action Dated: July 19, 20007

PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.114

process, starting a runtime environment in the hosting process, attaching a debugger to the hosting process, and preloading assemblies, before invoking the debugger, it is requested that the rejection of claims 1, 3-5, 7-18, 20-22, 24-26, and 28-38, under 35 U.S.C. §102, be reconsidered and withdrawn.

DOCKET NO.: MSFT-1757
Application No.: 10/626,958
Office Action Dated: March 30, 2007
Advisory Action Dated: July 19, 20007

PATENT
REPLY FILED UNDER EXPEDITED
PROCEDURE PURSUANT TO
37 CFR § 1.114

CONCLUSION

In view of the foregoing arguments, remarks, and amendments, it is respectfully submitted that this application is in condition for allowance. Reconsideration of this application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow this application for any reason, the Examiner is encouraged to contact the undersigned attorney to discuss resolution of any remaining issues.

Date: September 28, 2007

/Joseph F. Oriti/
Joseph F. Oriti
Registration No. 47,835

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439